

A method to manage the co-evolution of Product and Organization architectures

G. HARMEL, E. BONJOUR, M. DULMET

Laboratoire d'Automatique de Besançon

UMR CNRS 65 96

ENSMM - Université de Franche-Comté - 25 rue Alain Savary - 25000 Besançon

gharmel@ens2m.fr ebonjour@ens2m.fr mdulmet@ens2m.fr

Abstract - Complex systems design and especially automotive design is facing continuous technological evolution that needs stronger integration. To tackle this issue, a method for modeling products and organizations architectures is presented. This method will be completed by a "management by uncertainty" tool to help managers anticipate the co-evolutions of project domains and properly structure these domains.

Keywords: product architecture, organization structure, DSM, competencies, uncertainty management.

I. INTRODUCTION

Nowadays, firms' survival depends on their ability to deal with the constant evolution of customers needs. The fast reactivity and adaptation of firms to this changing environment is strongly related to an advanced control of their internal mechanisms. Concerning product development situations, researchers [ALL, 77] [PIM, 94] have highlighted the interdependency between three domains which needs to be modeled. These relevant domains are: Product, Processes, and Organization. We assume organization as the domain of competencies and actors. This hypothesis is frequently taken in many papers dealing with Organizations restructuring [SOS, 03].

Product architecture choice depends on the innovation and standardization policy of the firm. Process architecture depends on the design methodology adopted and is constrained by the functional decomposition of the firm into departments and teams. The choice of organization architecture is related to the other domains and so it may depend on their constraints.

In order to represent the above mentioned synergy, many tools and models have been developed. Unfortunately, they capture only static and partial views of the New Product Development (NPD) situations.

That is why, we develop here a method using matrices for domains representation. This matrix tool makes possible representing all the interfaces existing in the domains structures.

The proposed method uses the management by uncertainty to model first the interdependences between elements and domains and second the propagation of

uncertainties between them and the co-evolution resulting of it.

Our goal is to give system project managers a method that jointly represent products, organizations and processes architectures and that propagates constraints and evolution impacts through the three domains. Thus, this method is useful during the preliminary design phases where domains structure can be source of innovation.

In this paper, only the evolution of the product and the organization structure will be treated. The processes domain, as for it, will be the object of future work.

The proposed method of "management by uncertainty" is applied on an automotive component development project. The project example is however simplified in order to limit its complexity.

We will, first, introduce the uncertainty management principals, then we will present the DSM tool and how it is used to model the project domains architectures. Finally, we will deal with the application of the method of "management by uncertainty" on an automotive component development project. The project example is however simplified in order to limit its complexity.

II. UNCERTAINTY MANAGEMENT

The management by uncertainty covers several fields (project management, risk management, data processing, decision-making, mathematics...). This diversity makes existing approaches multiple and different. In the same way, methods objectives can vary from added value evaluation [BOU, 02] to uncertainty sources elimination [GOD, 96].

In this work, our approach is close to project management methods.

A. Uncertainty typology proposal

We assume that uncertainty is a cognitive concept relative to an observer/actor.

Loch and his coauthors [LOC, 00] model a development project not as a group of tasks but as a group of parameters (attributes) which influence the creation of value in the firm. Thus, they propose a typology composed of 5 points: complexity, variability, risks, ambiguity, and chaos.

In reference to Loch's typology, we suggest the following one in three points:

- Uncertainty by ambiguity: related to the existence or not of an element, a parameter or an entity (introduction of the element W in Fig. 1),
- Uncertainty by complexity: related to the existence or not of links, interactions between elements, parameters or entities (broken arrow between Y and X, X and W and W and Z in Fig. 1),
- Uncertainty by variability: related either to the fact of not taking into accounts some possible values of a parameter or to the fact of taking into account prohibited values (change on Z definition domain in Fig. 1).

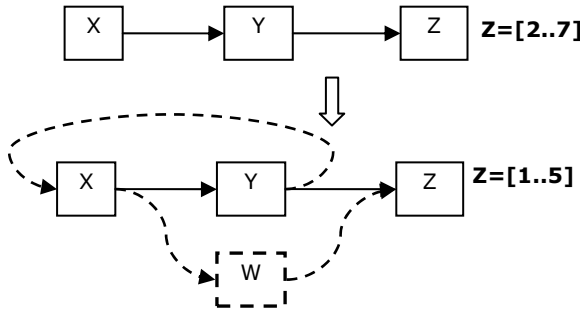


Fig. 1. Uncertainty typology application

Starting from an initial representation, Fig. 1 shows a complete representation in which the 3 types of uncertainties are identified.

B. Management by uncertainty

In this paper, is presented on the basis of an example a methodology of management by uncertainty. However, and that will be the subject of our future work, a data-processing tool based on our method can be developed to simulate and optimize the design situation.

Once realized, this tool can help project managers simulate and control project development evolution and specially the co-evolution of domains structures.

III. PRODUCT AND ORGANIZATION ARCHITECTURES MODELING

In the following paragraph, we introduce the Design Structure Matrix tool (DSM). This tool is at the basis of our model representing the co evolution of product and organization architectures.

A. Modeling by DSM

The design structure matrix (DSM) is becoming a popular modeling and analysis tool for system modeling, especially for purposes of decomposition and integration.

A DSM displays the relationships between system components in a compact and visual format. A DSM is a square matrix with identical row and column labels. In the example DSM in Fig. 1, elements along the diagonal have no sense. An off-diagonal mark signifies the dependency of one element on another. Reading across a row reveals what other elements the element in that row provides to; scanning down a column reveals what other elements the element in that column depends on. That is, reading down a column reveals input sources, while reading across a row indicates output sinks. Thus, in Fig. 2, element D provides something to elements C, E and G, and it depends on something from element E.

A more detailed overview of the DSM method is present in [STE, 81] and [EPP, 94].

	A	B	C	D	E	F	G
A	A	■					■
B	■	B					■
C	■		C				
D				D	■		
E				■	E	■	■
F					■	F	
G		■		■		■	G

Fig. 2. DSM example

Here a limited listing of the literature dealing with DSMs by domain:

- Product [PIMM, 94] [VAN, 01] [CHO, 01]
- Organization [MCC, 93] [SOS, 03]
- Processes [STE 81] [YAS, 99] [BRO, 02].

There are two main categories of DSMs: static and time-based. Static DSMs represent system elements existing simultaneously, such as components of a product structure or groups in an organization. Static DSMs are usually analyzed with clustering algorithms. In time-based DSMs, the ordering of the rows and columns indicates a flow through time: upstream activities in a process precede downstream activities, and terms like “feedforward” and “feedback” become meaningful when referring to interfaces. Time-based DSMs are typically analyzed using sequencing algorithms.

In this article, we straddle these various fields to deal with the product and the organization. It is assumed that the structure of the matrices have been already optimized by clustering algorithms.

B. Modular and integrative architecture

We are interested in this work in complex systems design (aircraft, cars, personnel computers ...), which often follows systemic engineering steps. Systemic exploration

[LEM, 99] is the principal tool of systemic engineering, it enables us to model the system by:

- Decomposing system into subsystems and subsystems into components [ALE, 64];
- Identifying the interactions which links all the components and makes system integrity;
- Immersing the system in an environment which explicit system's objectives.

Ulrich [ULR, 95] defines product architectures as “the scheme by which the function of a product is allocated to physical components.” A key feature of product architecture is the degree to which it is modular or integrative. In modular architectures, functional models of the product map one-to-one to its physical components. On the other hand, in integrative architectures, a large subset of functional models map to a single or small number of components.

Other researchers have investigated the implications of product modularity on various aspects of product development. Newcomb [NEW, 98] studied the effects of product modularity on product life cycle. Other researchers have studied the relation between product architecture and product portfolio definition [YU, 99]. The link between product architecture and supply chain has been addressed in the operations and management science literature [LEE, 96] [GUP, 99]. Few other researchers have studied how product modularity may affect testing strategies of design alternatives [THO, 98] [LOCH, 98].

In the engineering design field a large stream of research has focused on methods and rules to map functional models to physical components. Researchers have developed several architecting rules to map function to physical modules [SHA, 89] [LIU, 99] [STO, 00]. Other approaches view the functional model of a system as being described by an abstract functional decomposition that may, but do not need to, have a direct mapping onto physical decomposition of assemblies and subassemblies [BAX, 94].

Ulrich and Eppinger [ULR, 00] claim that the product architecture is also the scheme by which “the chunks of a product interact.” In complex products, the chunks of physical components are complex systems as well. They also argue that the challenge of establishing the architecture of these systems “is essentially identical to the architectural challenge posed at the level of the entire product.”

By using established concepts in the current product architecture literature we can categorize systems as modular or integrative based on how their corresponding components share design interfaces -within the system.

However, we also need to categorize systems according to how they share design interfaces with other systems as a result of the product architecture. Hence, at the system

level in order to define system modularity we need to specify whether we are looking at the system internally, as an independent entity, or externally, as an entity that interacts with other systems comprising the product.

We introduce here the concepts of modular and integrative systems, from an external perspective, that is, based on the existence of design interfaces between components of the same product that belong to different systems. We define modular systems as those whose design interfaces with other systems are clustered among a few physically adjacent systems, whereas integrative systems are those whose design interfaces span all or most of the systems that comprise the product due to their physically distributed or functionally integrative nature throughout the product.

C. An example of product DSM

Starting from Ulrich's architecture typology and recursivity concept, we can model system architecture at each level of decomposition as being composed of modular and integrative components. Fig. 3 illustrates this concept on a car, the engine subsystem is modeled as a single component in the first DSM, but we can apply system decomposition to the engine, we obtain the second DSM where we can identify main engine submodules as engine block, ignition ... We can notice that the same component can belong to two different modules as shown by the overlapping between intake and ignition modules [MCC, 93].

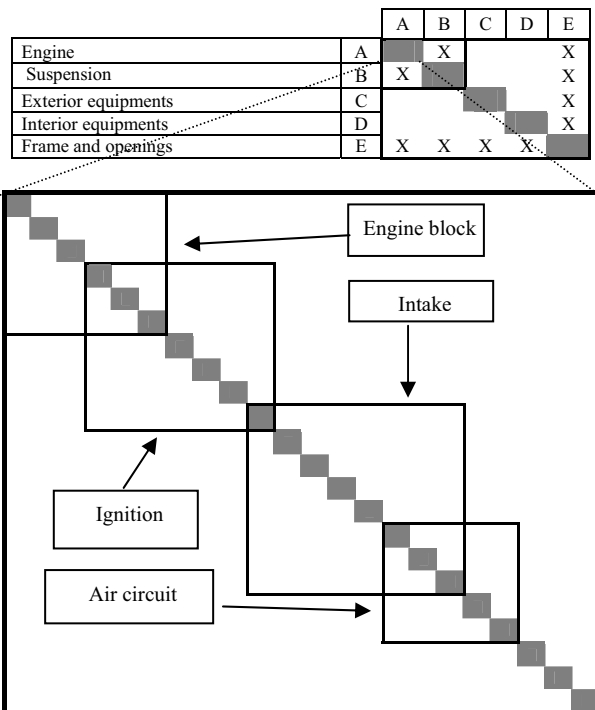


Fig. 3. Car product DSM [MCC, 93]

D. Product development organization structure

The product development organization is composed of all the actors participating to the design process. We model such organization structure with a DSM. We obtain so a matrix with actors in lines and columns, each matrix element represents an interaction between two actors [SOS, 02] [SOS, 03]. The interactions reported in the organization DSM are obtained through the documentation of technical communications occurring between actors in coordination-type interactions and especially by addressing their criticality and frequency.

IV. PRODUCT AND ORGANIZATION STRUCTURE CO-EVOLUTION

Elements composing a product module are strongly coupled. In fact, there is a strong space and exchange dependency between these elements. However, and by definition of a modular architecture, the modules themselves are slightly coupled. Thus, actors participating to the design of the same product module have frequent communications and exchange a great volume of information. On the other hand, two actors belonging to two different modules design components do not need to communicate because their components do not share interfaces.

Several researches on information theories and human communication [SOS, 02] [ALL, 97], have shown that communication quality depends on which communication media is used and on the distance between the transmitter and the receiver. Morelli [MOR, 95] concluded that in order to optimize the product development performance, actors designing components belonging to the same product module need to be regrouped in a single team where they share the same work space, the same media and the same information sources.

The influence of the product structure on organization one was formalized by Sosa [SOS, 03]. Our purpose in this work is to study this co-evolution by highlighting the actors' cognitive dimension.

A. Example presentation

Pimmler [PIMM, 94], Eppinger [EPP, 94] and Browning [BRO, 01] have focused in their researches on car design at Ford and General Motors companies. According to our own experience at a French automobile company, we find that cars are a good example to study the development situation of a complex product.

Cars are complex systems embedding an increasing number of "high-tech" components (e.g. Stability controllers, calculators, speed governor...). In order to overcome such system complexity, cars are modeled using both Ulrich structure typology and recursivity

concept. Fig. 3 and 4 show the structure representation of the car product and its development organization.

In the following part of this paragraph, the product example treated can be considered as being a generic car subsystem. The example starting point is drawn from an instant image of a car component (gearbox) development situation. Thus, we are not interested in identifying the mechanisms and decisions leading to this starting point situation.

		A	B	C	D	E
Engine Team	A		X			X
Suspension Team	B	X				X
Exterior equipments Team	C					X
Interior equipments Team	D					X
Frame and openings Team	E	X	X	X	X	

Fig. 4. Car development organization DSM

The generic subsystem has a hybrid structure (Fig. 5): 2 modules S1 (A, B, C, D) and S2 (E, F, G, H) and one integrative component I. We assume that each component is developed by a single actor -this assumption doesn't interfere with the need of actors to collaborate during the project, but the collaborative work will be related to the design for an example of a product module (S1 in Fig 1). Then, the development team has the same structure than the product – by extension of Morelli conclusions [MOR, 95]. We obtain so two modular teams, E1 (T1, T2, T3, T4) and E2 (T5, T6, T7, T8), and an integrative actor, T9 (Fig. 6).

		A	B	C	D	E	F	G	H	I
S1	A		X		X					
	B	X		X						X
	C		X		X					
	D	X		X		X				X
S2	E				X		X	X		X
	F					X			X	
	G					X			X	
	H						X	X		X
	I		X		X	X			X	

Fig. 5. Initial product DSM

		T1	T2	T3	T4	T5	T6	T7	T8	T9
E1	T1		X		X					
	T2	X		X						X
	T3		X		X					
	T4	X		X		X				X
E2	T5				X		X	X		X
	T6					X			X	
	T7					X			X	
	T8						X	X		X
	T9		X		X	X			X	

Fig. 6. Initial organization DSM

B. Product evolution situation

The call for technology innovation is sustained by strong customers' expectations in terms of performance, especially, reliability and safety. Moreover, the firm environment is characterized by a cut-throat competition. The pooling of all these facts makes the company

technological innovation dynamics faster than competences acquisition ones.

We propose to study different scenarios for the introduction of a new technology and its impacts on the co-evolution of the product and the organization structures. This example will be treated also under the angle of the "management by uncertainty" methodology.

The question which worries NPD managers when dealing with the introduction of a new technology is about the margin to leave to self-organization, whether it is necessary to impose a new organization and which one?

We will introduce various scenarios of possible evolutions of the product and its related development organization. DSM tool will help us represent the different structures for these domains.

The introduction of high-tech components into the studied product is materialized by the replacement of components D and E which are initially mechanical by electronic ones (D' and E'). This innovation is close to the case of replacement of jacks by electrical actuators.

The product structure illustrated by Fig. 7 shows the transitional product architecture resulting from the introduction of components D' and E', we notice that:

- We make the hypothesis that the product conserves its initial structure. The components D' and E' have replaced components D and E by occupying the same positions in the modules. Thus, component D' belongs to module S1 and E' to S2.
- Some of the interfaces linking new components D' and E' to product initial components become weakly specified. This is principally related to the fact that the introduction of new components in the product creates often new interfaces totally different from preexistent ones with new constraints and new parameters. The interfaces miss-defined in our example are: (C - D') and (G - E').
- There are new interfaces which are not altered by the introduction of E' and D'. This can be explained by the fact that either no interface specification modification is needed or the adaptation is already included in a component specification.

"Management by uncertainty" interpretation

By applying the "management by uncertainty" to this initial step of the evolving development product situation, we notice that the introduction of components E' and D' is in fact an exploration of "uncertainty by ambiguity". Knowing that "uncertainty by ambiguity" is always associated to "uncertainty by complexity", we can predict that the following steps will be based on the exploration of the impacts of introducing E' and D' in the product and in the organization structure.

First, when applying "uncertainty by complexity" exploration in product domain, we identify new interfaces

which need to be specified. These interfaces are (C - D') and (G - E').

		A	B	C	D'	E'	F	G	H	I
S1	A		X		X					
	B	X		X						X
	C		X		O					
	D'	X				X				X
S2	E'				X		X	O		X
	F					X			X	
	G					O			X	
	H						X	X		X
	I		X		X	X			X	

Fig. 7. Transient product DSM

Second, we have to associate to this product DSM an organization one. We identify so two possible scenarios for the organization structure.

In fact, the introduction of E' and D' is associated to the introduction into the product development team of two new actors T4' and T5'. These actors are in charge respectively of E' and D' design. T4' and T5' have principally competencies in electronics and automatics. The two scenarios identified are based on the organization structure. We have identified two possible forms (Fig. 8 and 9). The forms can be either dictated by the project manager during project planning or simply resulting from actors' self-organization during the project.

First scenario: development organization evolution

		T1	T2	T3	T4'	T5'	T6	T7	T8	T9
E1	T1		X		X					
	T2	X		X						X
	T3		X		-					
	T4'	X		-		X				X
E2	T5'				X		X	-		X
	T6					X			X	
	T7					-			X	
	T8						X	X		X
	T9		X		X	X			X	

Fig. 8. Transient organization DSM (1)

We notice that:

- The development organization has conserved its initial structure. The actors T4' and T5' belong respectively to E1 and E2 teams.
- Some actors' interfaces are lacking. This is due principally to the fact that the new teams are heterogeneous; actors do not share the same knowledge and the same communication media. We can imagine that communication interfaces between actors T4' and T3, and between T5' and T6 are inexistent especially because a collaborative work is necessary in order to design the corresponding product interfaces.
- The structure of the development organization presented here support competences diversification by putting together people having different competencies.

Second scenario: development organization evolution

		T1	T2	T3	T4'	T5'	T6	T7	T8	T9
E 1	T1		X							
	T2	X		X						X
	T3		X							
E 3	T4'	X				X				X
	T5'				X		X	-		X
E 2	T6					X			X	
	T7							X		
	T8						X	X		X
	T9		X		X	X			X	

Fig. 9. Transient organization DSM (2)

We notice that:

- The development organization has a different structure by comparison to the initial one. The actors T4' and T5' form a new team E3. The project teams are determined according to actors' principal competencies.
- The same actors' interfaces are lacking. Besides the reason mentioned in scenario one, we can add an other reason linked to the fact that an actor from team E3 and an actor from team E1 or E2 do not share the same working space. All these facts together do not facilitate the set-up of a collaborative working situation.
- The structure of the development organization presented here support competences improvement and knowledge capitalization, by putting together people having the same competencies.

"Management by uncertainty" interpretation

Uncertainty propagation from product domain to organization one allows us to apply "uncertainty by ambiguity" exploration on organization structure. This leads us to identify actors T4' and T5'.

The "uncertainty by complexity" exploration reveals new actors interfaces which needs to be specified. These interfaces are (T4'- T3) and (T5'- T6).

The satisfying structure of the development organization cannot yet be identified without characterizing the lacking interfaces. At this point, there is no difference between the two development organizations structures presented, they can only help us guessing the evolution way for each case.

C. Recovering optimal development situation

Evolving from the transient states presented in the previously identified scenarios to a satisfying development situation implies completing product and organization interfaces characterization.

This final step is equivalent for the management by uncertainty to exploring the uncertainties by variability. In fact, at this point, the miss-specified interfaces are identified. We need now to eliminate uncertainties

inherent to interfaces specifications by fixing all the parameters characterizing them.

From each scenario we can deduce a different possible final development situation

First scenario: back to initial state

Overcoming organization interfaces specification problem goes by evolving actors' competencies. Since actor T4' respectively actor T5' belongs to team E1 respectively E2, they need to share with their colleagues the same competencies and the same communication media. We can imagine that the aimed competencies evolution mode is so that each actor makes an effort to create a collective team competence. So, actors T1, T2, T3, T6, T7 and T8 have to acquire competencies in electronics and automatics while actors T4' and T5' have to acquire competencies in mechanics.

The resulting DSMs for product and organization are the same than the initial DSMs in Fig. 5 and 6.

Second scenario: towards an integrative structure

In this scenario, the actors' interfaces having specification problem link actors belonging to different teams. We can admit that unlike the precedent scenario it is more difficult to develop a collective competence between all the teams. In that case, one team has to carry out the competencies evolution needed. Since E3 team is the smallest and the newest one, actors T4' and T5' have to acquire mechanical competencies.

In that case, actors T4' and T5' become integrative actors sharing multiple competencies with different actors belonging to different teams. We obtain so an integrative development organization structure with three integrative actors (Fig. 11).

The modification in organization structure impacts product structure, component D' and E' become integrative. In fact, if the actors T4' and T5' carry out competencies shifting, then the components D' and E' have to carry out the adaptations needed in the corresponding interfaces specification.

Thus we obtain an integrative product structure as shown in Fig. 10.

		A	B	C	F	G	H	D'	E'	I
S1	A		X					X		
	B	X		X						X
	C		X					X		
S2	F					X			X	
	G						X		X	
	H				X	X				X
	D'	X		X					X	X
	E'				X	X		X		X
	I		X				X	X	X	

Fig. 10. Integrative product DSM

		T1	T2	T3	T6	T7	T8	T4'	T5'	T9
E1	T1		X					X		
	T2	X		X						X
	T3		X					X		
E2	T6					X			X	
	T7					X			X	
	T8				X	X				X
	T4'	X		X					X	X
	T5'				X	X		X		X
	T9		X				X	X	X	

Fig. 11. Integrative organization DSM

D. Co-evolution management

Without expressing any qualitative judgment on scenarios characteristics, we can admit that the manager choice between these two projects structures will be based on the following points:

- Modules robustness: if the firm design culture is based on the reuse of products components and modules, then the first scenario guarantees initial modules preservation, at the price of making the majority of actors evolving their competencies.
- Competencies dynamics: the cost of competencies evolution depends on many parameters and it varies from an actor to another. Thus, making electronic competency actors evolve towards mecatronics ones hides a manager conviction that this scenario is cheaper or quicker to realize.
- Integrative elements capacity: it is possible to make the integrative components integrate the adaptations needed and the integrative actors shift their competencies. But to which extent? Is it possible in future work to make the same choices?
- Innovation requirements: the strategic management has to choose between creating multi-competencies actors without any specialization and preserving specialized competencies with taking the risk of facing problems in integrating new competencies.

V. DISCUSSION ON MANAGEMENT BY UNCERTAINTY IMPLEMENTING

The "management by uncertainty" methodology presented in this paper can be improved in order to make it easier to use and better formalized. The aimed formalization goes through the adoption of a metric to evaluate uncertainty at different states (steps) of the development project evolution.

This metric is a way to characterize from uncertainty point of view the interfaces presented by DSM tool and so serves at evaluating uncertainty by variability.

Product interface uncertainty can be formalized as follows:

- Associate to each product interface a set of objective design parameters;

- Uncertainty evaluation takes into account the real set of parameters addressed and the variability associated to each parameter;
- Associate to each interface a degree of criticality based on its uncertainty value and a degree of importance (number of parameters, number of product and organization elements impacted...)

Development organization interface uncertainty can be formalized as follow:

- Associate to each actors' interface a set of collaboration and communication parameters.
- Uncertainty is evaluated through a scoring (by experts or by the project manager) of the previously identified parameters

In addition of the metric adopted, a mapping between parameters and elements is necessary in order to propagate the uncertainties inside and outside the domains.

Through the different points proposed in order to formalize the management by uncertainty, we can build a simulation tool serving at:

- Starting from an initial development situation with a fixed product and organization structure, we can simulate the effect of uncertainty introduction into the model.
- Uncertainty introduction can be either ambiguity type –as developed in the example treated, or complexity and even variability type.
- The expected output of this simulation tool is the appropriate product and organization structure for distributing, eliminating and absorbing the uncertainties introduced.

VI. CONCLUSIONS AND FUTURE WORK

Complex systems are difficult to model and to represent according to the great number of components and the exponential cardinal of links. However, Ulrich structure typology and the recursivity concept allow us to overcome these difficulties by making complex system modeling possible. The DSM tool as for it, allows displaying elements relationships in a compact and visual format, and so it overcomes complex systems representation difficulties.

In this paper, is presented, based on an example drawn from a real automotive development project, a methodology to manage and control the evolution of both the product architecture and the development organization structure.

This method uses the management by uncertainty to propagate the uncertainties in and between the product and the organization domains. It aims at helping the system project manager anticipate and predict the effects

of uncertainties introduction on the project structure evolution.

The method proposed needs to be completed in order to form a powerful tool for simulating project structure evolution. That's why, we aim at developing the following points in our future work:

- Introduce the processes domain in order to deal with the three faces of the development project (Product, organization and processes). Thus, we can model the simultaneous evolution of the three domains.
- Identify efficient domain structures for uncertainties elimination.
- Formalize the method as discussed in paragraph V. Thus, we can evaluate different scenarios of project structure evolution and control project uncertainties distribution.
- Qualify domains robustness against project uncertainties modification

REFERENCES

- [ALL, 77] T.J. Allen, Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization. MIT Press, Cambridge, MA, 1977.
- [ALL, 97] T. J. Allen, "Architecture and Communication Among Product Development Engineers." Sloan Working Paper, # 3983, MIT Sloan School of Management, 1997
- [ALE, 64] C. Alexander, Notes on the Synthesis of Form. Harvard University Press, Cambridge, MA, 1964.
- [BAX, 94] J. E. Baxter, N. P. Juster, and A. Pennington, "A Functional Data Model for Assemblies Used to Verify Product Design Specifications." *Proceedings IMechE, Part B*, **208**, pp. 235–244, 1994.
- [BOU, 02] S. Bougaret, "Prise en compte de l'incertitude dans la valorisation des projets de recherche et développement: la valeur de l'information nouvelle." Thèse de doctorat, Institut National Polytechnique de Toulouse, 2002.
- [BRO, 01] T. R. Browning, "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions." IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 48, NO. 3, 2001.
- [BRO, 02] T. R. Browning, "Process integration using the design structure matrix." Systems Engineering. Vol 5. No 3, 2002.
- [CHO, 01] S-H. Cho, "An Integrated Method for Managing Complex Engineering Projects Using the Design Structure Matrix and Advanced Simulation." Master of Science Thesis, MIT, 2001.
- [EPP, 94] S. D. Eppinger, D.E. Steven, R. P. Smith and D. A. Gebala, "A Model-Based Method for Organizing Tasks in Product Development." Research in Engineering Design, vol. 6, no. 1, pp. 1-13, 1994.
- [LEE, 96] H. L. Lee, "Effective Inventory and Service Management through Product and Process Re-Design." *Oper. Res.*, **44**, pp. 151–159, 1996.
- [LEM, 99] J.-L. Le Moigne, Modélisation des systèmes complexes, éd Dunod, 1999.
- [LIU, 99] Y. C. Liu, A. Chakrabarti, and T. P. Bligh, "Transforming Functional Solutions into Physical Solutions." Proceedings of the ASME Design Theory and Methodology Conference, DETC/DTM-8768, Las Vegas, NV, 1999.
- [LOC, 98] C. H. Loch and C. Terwiesch, "Communication and Uncertainty in Concurrent Engineering." *Management Science* / Vol 44, No. 8, 1998.
- [LOC, 00] C.H. Loch, M.T. Pich and A. De Meyer, "Project uncertainty and management styles." INSEAD R&D 2000/31/TM/CIMSO 10. Fontainebleau, France, 2000.
- [GUP, 99] S. Gupta, and V. Krishnan, "Integrated Component and Supplier Selection for a Product Family." *Productions and Operations Management*, **10**, pp. 291–308, 1999.
- [GOD, 96] P. Godfrey, "Control of risk: a guide to the systematic management of risk from construction." CIRIA, SP125, London, 1996.
- [MCC, 93] K. R. McCord and S. D. Eppinger, "Managing the Integration Problem in Concurrent Engineering, M.I.T. Sloan School of Management, Cambridge, MA, Working Paper no.3594, 1993.
- [MEN, 02] J-P. Meinadier, LE METIER D'INTEGRATION DE SYSTEMES, Hermes Science Publications, 2002.
- [MOR, 95] M. D. Morelli, S. D. Eppinger, and R. K. Gulati, "Predicting Technical Communication in Product Development Organizations." IEEE Transactions on Engineering Management, vol. 42, no. 3, pp.215–222, 1995.
- [PIM, 94] T. U. Pimmler, and S. D. Eppinger, "Integration Analysis of Product Decompositions." Proceedings of the ASME Design Theory and Methodology Conference, DE-Vol 68, pp. 343–351, 1994.
- [SHA, 89] V. Shapiro, and H. Voelcker, "On the Role of Geometry in Mechanical Design." *Res. Eng. Des.*, **1**, pp. 69–73, 1989.
- [SOS, 02] M. E. Sosa, S. D. Eppinger, M. Pich, D. G. McKendrick, and S. K. Stout, "Factors That Influence Technical Communication in Distributed Product Development: An Empirical Study in the Telecommunications Industry." IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT VOL. 49 NO. 1, 2002.
- [SOS, 03] M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "Identifying modular and integrative systems and their impact on design team interactions." ASME Journal of Mechanical Design, **125**, pp. 240–252, 2003.
- [STE, 81] D.V. Steward, "The Design Structure System: A Method for Managing the Design of Complex Systems." IEEE Transactions on Engineering Management, vol. EM-28, no 3, pp. 71–74, 1981.
- [STO, 00] R. B. Stone, , K. L. Wood and R. H. Crawford, "A Heuristic Method for Identifying Modules for Product Architectures." *Des. Stud.*, **21**, pp. 5–31, 2000.
- [THO, 98] S. H. Thomke, "Managing Experimentation in the Design of New Products." *Manage. Sci.*, **44**, pp. 743–762, 1998.
- [ULR, 95] K. T. Ulrich, "The Role of Product Architecture in the Manufacturing Firm." *Research Policy*, vol. 24, 1995, pp. 583–607.
- [ULR, 00] K. T. Ulrich and S. D. Eppinger, Product Design and Development, McGraw Hill, New York, 2000.
- [VAN, 01] M. Van Wie, J. Greer, M. Campbell, R. Stone and K. Wood, "Interfaces and Product Architecture." ASME Design Engineering Technical Conference Proceedings, DETC01/DTM-21689, 2001.
- [YAS, 99] A. Yassine, D. R. Falkenburg, and K. Chelst, "Engineering Design Management: An Information Structure Approach". International Journal of Production Research, vol. 37, no. 13, 1999, pp. 2957–2975.
- [YU, 99] J. S. Yu, J. P. Gonzalez-Zugasti and K. N. Otto, "Product Architecture Definition Based Upon Customer Demands." ASME J. Mech. Des., **121**, pp. 329–335, 1999.